

Logiciel Libre

Cours 5 — Economics

Stefano Zacchioli
zack@irif.fr

Laboratoire IRIF, Université Paris Diderot

2018-2019

URL <https://upsilon.cc/zack/teaching/1819/loglib/>
Copyright © 2014-2018 Stefano Zacchioli
© 2013-2014 Roberto Di Cosmo
© 2010-2012 Jesus M. Gonzalez-Barahona, Felipe Ortega
License Creative Commons Attribution-ShareAlike 4.0 International License
<https://creativecommons.org/licenses/by-sa/4.0/>



- 1 FOSS economy
- 2 FOSS business models
- 3 Recent developments

Libre n'est pas gratuit, et gratuit n'est pas libre

non libre, gratuit Internet Explorer, MacTCP, Acrobat Reader,
freeware, ...

Libre n'est pas gratuit, et gratuit n'est pas libre

non libre, gratuit Internet Explorer, MacTCP, Acrobat Reader,
freeware, ...

non libre, non gratuit le plus connu ...

Libre n'est pas gratuit, et gratuit n'est pas libre

non libre, gratuit Internet Explorer, MacTCP, Acrobat Reader, freeware, ...

non libre, non gratuit le plus connu ...

libre, gratuit Firefox, Linux, Gimp, LibreOffice, SciLab, OCaml, R, ...

Libre n'est pas gratuit, et gratuit n'est pas libre

non libre, gratuit Internet Explorer, MacTCP, Acrobat Reader, freeware, ...

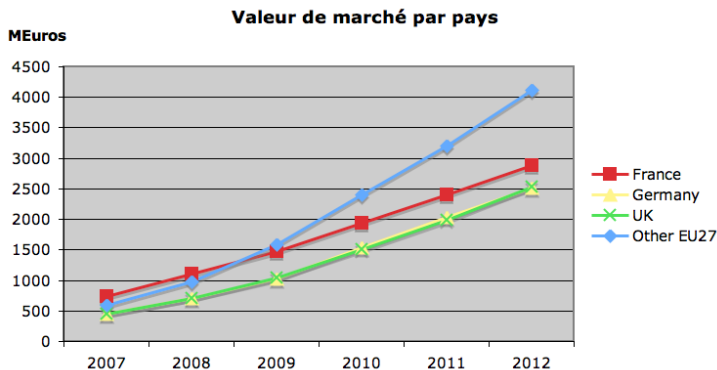
non libre, non gratuit le plus connu ...

libre, gratuit Firefox, Linux, Gimp, LibreOffice, SciLab, OCaml, R, ...

libre, non gratuit distributions GNU/Linux commerciales, développement de logiciel libre sur demande, ...

Why bother? — as future IT workers (redux)

- Free Software is everywhere
- participation in Free Software : a big plus on CVs
- in France : $\approx 10\%$ of IT GDP in 2014 (≈ 4 BEUR), constantly growing in spite of the crisis



More info :

<http://cn11.fr/news/etude-pac-logiciel-libre-2015-2020/>

Quelques repères

- en 2014, la filière Logiciel Libre française pèse plus de 4 milliards d'euros (PAC)
- la France a des belles “success stories” :
 - ▶ Videolan : logiciel multimédia plebiscité
 - ▶ JBoss (WildFly) : racheté 400M\$ par RedHat en 2006
 - ▶ eNovance
- RedHat
 - ▶ 1+ milliard de \$ de chiffre d'affaires depuis 2010 (2.9 en 2017)
 - ▶ racheté par IBM en 2018 (34 milliard \$)

Quelques repères

- en 2014, la filière Logiciel Libre française pèse plus de 4 milliards d'euros (PAC)
- la France a des belles “success stories” :
 - ▶ Videolan : logiciel multimédia plebiscité
 - ▶ JBoss (WildFly) : racheté 400M\$ par RedHat en 2006
 - ▶ eNovance
- RedHat
 - ▶ 1+ milliard de \$ de chiffre d'affaires depuis 2010 (2.9 en 2017)
 - ▶ racheté par IBM en 2018 (34 milliard \$)

Question naturelle

Une économie du *gratuit*?

- 1 FOSS economy
- 2 FOSS business models
- 3 Recent developments

Pour comprendre, il faut revenir aux définitions de base.

Definition (Économie)

L'étude de comment une société choisit d'utiliser des ressources *limitées* pour produire, échanger et consommer des biens et services. *Ruffin, Gregory, "Principles of Economics", 1990*

Logiciel Libre : les bases des modèles économiques

Pour comprendre, il faut revenir aux définitions de base.

Definition (Économie)

L'étude de comment une société choisit d'utiliser des ressources *limitées* pour produire, échanger et consommer des biens et services. *Ruffin, Gregory, "Principles of Economics", 1990*

Sans rareté de ressources, pas d'économie.

Logiciel Libre : les bases des modèles économiques

Pour comprendre, il faut revenir aux définitions de base.

Definition (Économie)

L'étude de comment une société choisit d'utiliser des ressources *limitées* pour produire, échanger et consommer des biens et services. *Ruffin, Gregory, "Principles of Economics", 1990*

Sans rareté de ressources, pas d'économie.

logiciel propriétaire : logique d'offre

se fonde sur la "rarété" artificielle des copies d'un logiciel ...

Logiciel Libre : les bases des modèles économiques

Pour comprendre, il faut revenir aux définitions de base.

Definition (Économie)

L'étude de comment une société choisit d'utiliser des ressources *limitées* pour produire, échanger et consommer des biens et services. *Ruffin, Gregory, "Principles of Economics", 1990*

Sans rareté de ressources, pas d'économie.

logiciel propriétaire : logique d'offre

se fonde sur la "rarété" artificielle des copies d'un logiciel ...

logiciel libre : logique de demande

supprime la rareté des copies, et révèle les autres ressources "rares"

Ressources rares : les compétences

C'est la base du modèle du *service* proné par Stallman depuis 1985.

Beaucoup de consultants individuels font du service.

Plusieurs sociétés de service en Logiciel Libre (SS2L) en France se basent sur ce modèle : Smile (la plus grande), Linagora, Alterway, en sont des exemples.

Economie du logiciel : licences vs. service

modèle centré sur les licences : profit non proportionnel au travail, peu ou pas d'emplois

modèle centré sur les services : tendance naturelle des grandes sociétés (IBM, Oracle etc.), profit proportionnel au travail, à la limite, le Logiciel Libre

(2002)

	Income	Profit	%	Employees	P/E^1	E/P^2
IBM	81,667 M\$	6,328 M\$	7	290.000		
Oracle	7,143 M\$	955 M\$	13	40.000		
Microsoft	20,000 M\$	8,000 M\$	40	29.000		

-
1. "taxe" collecté par employé
 2. employés à *votre* service

Economie du logiciel : licences vs. service

modèle centré sur les licences : profit non proportionnel au travail, peu ou pas d'emplois

modèle centré sur les services : tendance naturelle des grandes sociétés (IBM, Oracle etc.), profit proportionnel au travail, à la limite, le Logiciel Libre

(2002)

	Income	Profit	%	Employees	P/E^1	E/P^2
IBM	81,667 M\$	6,328 M\$	7	290.000	21 \$	45
Oracle	7,143 M\$	955 M\$	13	40.000	23 \$	41
Microsoft	20,000 M\$	8,000 M\$	40	29.000	275 \$	3

-
1. "taxe" collecté par employé
 2. employés à *votre* service

Evolution de la situation

2002

	Income	Profit	%	Employees	P/E	E/P
IBM	81,667 M\$	6,328 M\$	7	290.000	21 \$	45
Oracle	7,143 M\$	955 M\$	13	40.000	23 \$	41
Microsoft	20,000 M\$	8,000 M\$	40	29.000	275 \$	3

2006

	Income	Profit	%	Employees	P/E	E/P
IBM	91,000 M\$	8,000 M\$	9	366.500	21 \$	46
Oracle	14,000 M\$	3,300 M\$	24	56.000	58 \$	17
Microsoft	44,282 M\$	12,600 M\$	28	71.000	177 \$	6

2015

	Income	Profit	%	Employees	P/E	E/P
IBM	81,741 M\$	14,538 M\$	18	377.757	38 \$	26
Oracle	38,230 M\$	13,870 M\$	36	135.070	103 \$	10
Microsoft	93,580 M\$	18,160 M\$	19	118.584	153 \$	7

Ressources rares : l'infrastructure

- Exemple : les forges, comme SourceForge ou GitHub, qui fournissent un environnement de développement complet.
- Elles peuvent vendre des services à valeur ajoutée en valorisant une infrastructure largement connue.

- Un exemple essentiel : les distributions GNU/Linux, comme RedHat
- Un autre exemple : AdaCore.

Ressources rares : la communauté

- Un logiciel libre sans communauté a rarement beaucoup de valeur.
- L'accès à une communauté (decision, interaction, vision sur l'évolution technologique) est une ressource rare.
- Des développeurs phare sont recrutés par des entreprises en raison de leur compétences.
- Des marchés sont gagnés par des entreprises grace au fait qu'elles ont des développeurs phare.

Ressources rares : la communauté, bis

- Tout le monde peut faire une copie des sources d'un projet libre.
- Maintenir une version séparée d'un logiciel peut coûter très cher (il faut maintenir des centaines de patch)
- Très peu de personnes ont le droit de committer/merger dans la branche principale d'un logiciel libre !

Le cas des *éditeurs* de logiciels libres

- Une typologie particulière d'acteurs se développe beaucoup dans le monde du logiciel libre, celle des éditeurs de logiciel libre.

Le cas des *éditeurs* de logiciels libres

- Une typologie particulière d'acteurs se développe beaucoup dans le monde du logiciel libre, celle des éditeurs de logiciel libre.
- On parle d'éditeurs quand une partie significative des prérogatives de la communautés (droits de commit, développeurs phare, décision sur l'évolution du logiciel) se retrouve concentrée essentiellement dans les mains d'un seul acteur.

Le cas des *éditeurs* de logiciels libres

- Une typologie particulière d'acteurs se développe beaucoup dans le monde du logiciel libre, celle des éditeurs de logiciel libre.
- On parle d'éditeurs quand une partie significative des prérogatives de la communautés (droits de commit, développeurs phare, décision sur l'évolution du logiciel) se retrouve concentrée essentiellement dans les mains d'un seul acteur.
- Dans ce cas, l'éditeur devient un référent incontournable des grands utilisateurs du logiciel.

Le cas des *éditeurs* de logiciels libres

- Une typologie particulière d'acteurs se développe beaucoup dans le monde du logiciel libre, celle des éditeurs de logiciel libre.
- On parle d'éditeurs quand une partie significative des prérogatives de la communautés (droits de commit, développeurs phare, décision sur l'évolution du logiciel) se retrouve concentrée essentiellement dans les mains d'un seul acteur.
- Dans ce cas, l'éditeur devient un référent incontournable des grands utilisateurs du logiciel.
- Exemples : RedHat (qui a racheté JBoss), MySQL (avant rachat par Oracle), Nuxeo, Nexedi, Talend, Qualixo, Engineering, Mozilla, XWiki, AdaCore, etc.

Un temps d'apprentissage

- Cette rareté n'est pas toujours perçue clairement : jouer le *free rider* est une stratégie qui peut rapporter à court terme !
- Mais cela comporte des risques lourds à moyen terme
- Une fois le message compris, les flux financiers entre revendeurs/intégrateurs, et les vrais développeurs des logiciels libres s'établissent naturellement.

Ressource rares : le droit de changer la licence

- Nous avons vu que qui detient le droit sur un logiciel peut l'offrir sous des licences différentes à des acteurs différents.
- Cela fait de cet acteur un partenaire incontournable pour toute une classe de clients, et ouvre la voie à des modèles économiques qui génèrent des revenus á partir de qui veut des versions du même code sous conditions differentes

Outline

1 FOSS economy

2 FOSS business models

3 Recent developments

FOSS *per se* is not, and it has never been,
a business model.

- Companies must design a **strategy**, create a **business plan** and ensure **securing benefits** aimed to a sustainable growth.
- FOSS can improve **viability** and **efficiency** with many business models.

<http://smeguide.conecta.it/>
<http://smeguide.conecta.it/FLOSSguide.pdf>

- Report elaborated for FOSSMETRICS, EU FP6 project.
- Analysis of 218 companies receiving at least 25% of their total revenues directly or indirectly from FOSS.
- Identifying common business strategies around FOSS.
- Recommendations about required conditions to apply each model.

Axes influencing software landscape

- **Licensing model**
 - ▶ Proprietary vs. FOSS.
- **Development model**
 - ▶ Barriers to collaboration.
 - ▶ Single developer/reduced group vs. large community, global outreach.
- **Business model**
 - ▶ Type of revenues model.
 - ▶ Numerous options: Training, support, on-demand changes, productizing, SaaS, etc.

Strategic use of FOSS

- The Gartner consulting group pointed out: in 2012, use of libre software has reached 90% of companies worldwide.
- Companies forced to adapt their software to work under multiple conditions.
- Emerging trend in big companies towards source code release under FOSS licenses. More active interaction with nearby projects.
- Proprietary software success is now guaranteed only if there are no alternative and reliable FOSS options.

FLOSSMETRICS taxonomy

- 1 **Dual licensing**: FOSS version and proprietary version.
- 2 **Open core**: Allows mixing FOSS and proprietary elements.
- 3 **Product specialists**: Superior knowledge, additional services.
- 4 **Platform providers**: Integration, product testing.
- 5 **Aggregate support providers**: One-stop support for multiple FOSS products.
- 6 **Selection/consulting companies**: Closer to the analyst role, minimum impact on FOSS communities.
- 7 **Legal certification and consulting**: Assessment on license compatibility.
- 8 **Training and documentation**: Either as part of a broader support contract or companies exclusively devoted to this market area.
- 9 **R&D cost sharing**: Initial investment + creating community to reduce R&D costs.
- 10 **Indirect revenues**: Baseline for sales of associated products or services (commodities).

Dual (or multi) licensing

- Examples: MySQL, Berkeley DB, Asterisk
- The copyright owner selects a multi-licensing strategy.
 - ▶ **Copyleft license**: To obtain some benefits (community contribution, integration in other libre software, etc.).
 - ▶ **Proprietary license or permissive free license**: Other companies might integrate this software with proprietary software.
- Leverage adoption and visibility, offering alternatives to proprietary software companies.
- But it may also limit contributions from FOSS developers, who are forced to release copyright ownership—or equivalent—to ensure the company has the right to dual license.
 - ▶ see the [CAA/CLA debate](#)

Open core / freemium

- Also based on dual licensing, but this is not the business foundation.
- Examples: GitLab, Neo4j, Redis
- Different features linked to licenses:
 - ▶ **FOSS core**: Basic version of the software, including minimal working functionalities, is FOSS.
 - ▶ **Proprietary added-value functionalities**: Most sophisticated features, targeting the enterprise sector and expert audience, are released under a payware, proprietary license.
- Revenues concentrate on commercialization of proprietary components, plus support and extension/customization.
- FOSS developers might distrust the project, for the same reasons as dual licensing.

Product specialists

- Company offering or creating maintenance for a specific software product, usually in a concrete market.
- Example: Alfresco (Content Management System).
- Main revenues come from training, consulting, specialized support for installation, deployment and configuration, extensions, customization, etc.
- Typical model “best product knowledge here” or “best source code here”.
- It is mandatory to retain the best team of specialists.
- Main risk: other competitors entering at the same level of knowledge and specialization in the same market.

Platform providers

- Selection, support, integration and services around a set of projects integrated in a single, tested and verified product.
- Key points:
 - ▶ Verification process to meet quality standards.
 - ▶ Additional services (support, training, customization).
- Example: RedHat.
- The source code is FOSS, but the product name and logo are trademarks.
- Considerable effort might be needed to eliminate them from files, documentation, etc.

Aggregate support providers

- Centralized support point for different libre software products.
- Two strategies:
 - ▶ Employing developers specialized in those products.
 - ▶ Redirecting petitions to other support specialists.
- Clear benefit for large projects whose costs could raise due to excessive diversification of support channels (comprehensive help-desk).
- It only works for clients with many different products running on mission-critical platforms (e.g., large public administration deployments). They assure exclusive support contracts.
- Example: OpenLogic.

Consulting — component selection

- Usually, they do not develop software.
- Assessment about selection and evaluation of software products and services.
- It might save costs for clients (software selection process can add up to 45% of total costs of software adoption).
- Experience to undertake selection process is not easily replicable.
- Sometimes, they develop dedicated tools aimed for selection, that they do not release as free software to stop competition.
- Example: Open WebApps.

- They do not usually develop software, either.
- Legal assessment for companies about complex problems derived from multiplicity of libre software licenses and their compatibilities.
- They might save costs for integrators less familiarized with selection process and licensing checking.
- Example: Palamida, BlackDuck, Antelink.

Training and documentation

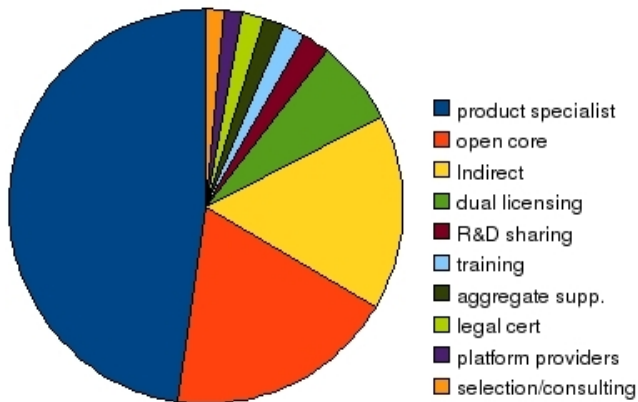
- Focused on important aspects of libre software, but not directly related to development.
 - ▶ Training users and tech staff.
 - ▶ Support for certification processes (technical staff and end-users).
 - ▶ Creation of high quality documentation.
- It is very helpful if supported products have a large user base (size of target market).
- Certification programs might provide extra income, conditions for collaboration agreements (partnerships).
- Example: Gbdirect.

- Company interested in software development decides to release the code to benefit from contributions from large community of interested developers.
- Development cost of closed source alternative would be prohibitive.
- Viability conditions:
 - ▶ Great branding and public image to attract attention from stakeholders.
 - ▶ Solvency to develop and nurture community participation.
 - ▶ Investment in project support (IDE, events, promotion, etc).
- Example: modern trade association centered around FOSS products

Indirect revenues

- Companies that decide to develop libre software or funding libre software projects to leverage their business on related products.
- Some examples:
 - ▶ **Hardware makers:** Intel, Dell, etc. broaden their target audience base providing better support for libre software platforms.
 - ▶ **Tech books publishers, event organizers:** O'Reilly
- Stakeholders that indirectly benefit from success of libre software projects.

Distribution of models identified in the study



Evolution CAGR in FOSS business

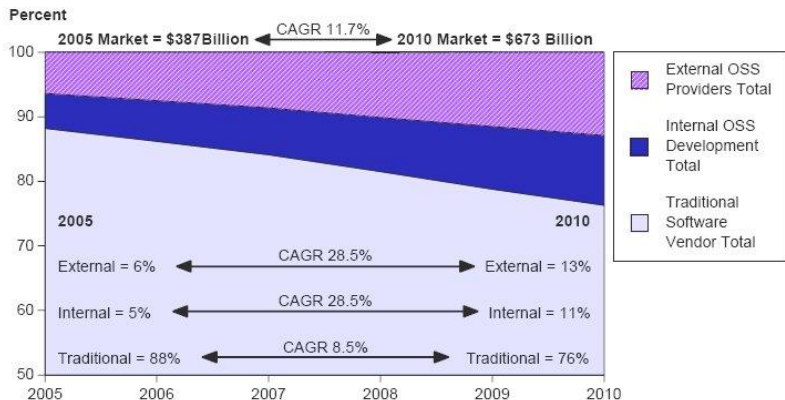


Figure – Taken from “Open Source going mainstream”, Gartner group report.

CAGR: Compound Annual Growth Rate

Companies running business around FOSS (selection)



Figure – Taken from “FOSS: A guide for SMEs”, Carlo Daffara (FOSSMETRICS EU FP6)

Outline

1 FOSS economy

2 FOSS business models

3 Recent developments

Crowdfunding

Pre-order / reverse-bounty

- one-shot fundraising, e.g., for a specific release
- Kickstarter, Indiegogo, etc.
- Examples: LiveCode (500 kGBP), Font Awesome (1 M\$), Diaspora (200 k\$), Ghost (200 kGBP)
- Problem: long term sustainability

Stable income

- monthly/recurrent payment, subsidized by the community
- Patreon, Snowdrift
- Examples: Mastodon (4.5 k\$/month)

Core Infrastructure Initiative

<https://www.coreinfrastructure.org>

Funded projects (excerpt):

- GnuPG
- OpenSSH
- OpenSSL
- Reproducible Builds

Tidelift

<https://tidelift.com>

Covered projects (excerpt):

- Apache Struts
- Babel.js
- Beautiful Soup
- Celery
- CherryPy
- Material-UI
- Paramiko
- Urllib3
- Vue.js

References

- “FLOSS: A guide for SMEs”. <http://smeguide.conecta.it/>
- “The Magic Cauldron”, Eric Raymond. <http://www.catb.org/esr/writings/magic-cauldron/magic-cauldron.html>
- “Selling Free Software”,
<https://www.gnu.org/philosophy/selling.html>
- “Selling [License] Exceptions”, <https://www.gnu.org/philosophy/selling-exceptions.html>
- “Livre blanc sur les modèles économiques du logiciel libre”, April, <http://www.april.org/livre-blanc-sur-les-modeles-economiques-du-logiciel-l>